



# Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Installation Manual

*Egon Kuster*

**Command and Control Division  
Information Sciences Laboratory**

DSTO-GD-0378

## **ABSTRACT**

The Web Data Collection Tool (WDCT) allows for the collection of demonstration assessment information from geographically disperse locations. This installation manual provides the WDCT administrator with instructions on how to install and configure the WDCT software. The WDCT in the past has been used for the assessment of JWID (Joint Warrior Interoperability Demonstration) and the CINC21 (Commander in Chief for the 21st Century) ACTD (Advanced Concept Technology Demonstrator). WDCT is a web-based tool that connects to an Oracle database to store its analysis information and configuration settings. Over the past few years that the WDCT has been used within JWID it has proven to be a valuable tool in eliciting feedback from the warfighters using the demonstrations being assessed.

## **RELEASE LIMITATION**

*Approved for public release*

*Published by*

*DSTO Information Sciences Laboratory  
PO Box 1500  
Edinburgh South Australia 5111 Australia*

*Telephone: (08) 8259 5555*

*Fax: (08) 8259 6567*

*© Commonwealth of Australia 2003*

*AR-012-921*

*September 2003*

**APPROVED FOR PUBLIC RELEASE**

# Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Installation Manual

## Executive Summary

The Web Data Collection Tool was developed to help analysts collect and analyse qualitative data about demonstrations within JWID (Joint Warrior Interoperability Demonstration). The WDCT uses both Java technology and database technology to capture analysis information from the warfighters using a web-based interface.

There are essentially two parts to the WDCT system. The first being the Java Servlets and Java Server Pages (JSP) that present the HTML based web interface. The second is information entered into the HTML forms is then sent back to the server and stored in an Oracle database.

All demonstrations, questions and answers are stored in the Oracle database. The demonstrations and questions can be edited via the web interface and from any computer that is connected to the same network. Reports about the stored data can also be produced and displayed in the same web interface.

This manual describes the installation process for installing each WDCT component. It is expected that the person installing the software is experienced with using Oracle and Java, although this is not mandatory. This manual also contains a description of how to install multiple instances of the WDCT onto a single machine.

The content of this manual is an invaluable resource for all who are responsible for maintaining or developing the WDCT. For further information about how to use the WDCT or develop the WDCT, refer to the User Guide (DSTO-GD-0383) and Developer Guide (DSTO-GD-0377).

# Author

## **Egon Kuster**

Command and Control Division

*Egon Kuster as worked as a web developer for the past six years. Current work is as the Australian Coalition Theatre Logistics (CTL) Technical Manager overseeing the research and development into the CTL demonstrator, which is being jointly developed between Australia and America. Egon has also been the South Australian node manager of the Combined Federated Battle Laboratory network (CFBLnet), which is a large multinational secret demonstration network. Primary skills include the development of large, complex distributed solutions based on web technologies. Egon has experience in creating software solutions to information management, distributed data access, Enterprise Information Portals (EIP) and web service communication.*

---

# Contents

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>1.1. JWID Description .....</b>	<b>1</b>
<b>1.2. JWID Assessment Process Description .....</b>	<b>2</b>
<b>1.3. WDCT Description .....</b>	<b>3</b>
 <b>2. INSTALLATION .....</b>	 <b>4</b>
<b>2.1. Installation Introduction .....</b>	<b>4</b>
<b>2.2. Installation Procedure .....</b>	<b>5</b>
2.2.1. Java .....	5
2.2.1.1. To install Java .....	5
2.2.2. Oracle .....	5
2.2.2.1. Install Oracle 8i.1.7 Software .....	6
2.2.2.2. Install Oracle 9i Software .....	6
2.2.2.3. Disable the Oracle HTTP Server .....	7
2.2.2.4. Configuring Oracle .....	8
2.2.2.5. Create WDCT Tables and Populate Initial Values .....	9
2.2.3. Jakarta Tomcat .....	10
2.2.3.1. Install WinZip .....	10
2.2.3.2. Install Jakarta Tomcat .....	11
2.2.3.3. Set up Jakarta Tomcat Environment Variables .....	11
2.2.4. WDCT Files .....	13
2.2.5. Configure Components .....	13
2.2.5.1. Edit Configuration Files .....	14
2.2.5.2. Oracle JDBC drivers for Jakarta Tomcat: .....	15
2.2.5.3. Set Up Jakarta Tomcat To Run As A Service: .....	15
2.2.6. Microsoft Internet Explorer 5.5 .....	16
 <b>3. MULTIPLE INSTANCES .....</b>	 <b>17</b>
<b>3.1. Jakarta Tomcat .....</b>	<b>17</b>
<b>3.2. Oracle .....</b>	<b>18</b>
<b>3.3. Settings .....</b>	<b>19</b>
 <b>4. TROUBLESHOOTING .....</b>	 <b>19</b>
 <b>APPENDIX A: SERVER.XML LISTING .....</b>	 <b>21</b>
 <b>APPENDIX B: WEB.XML LISTING .....</b>	 <b>26</b>

# Abbreviations

ADF	Australian Defence Force
API	Application Programming Interface
C4ISR	C4 Intelligence Surveillance and Reconnaissance
C4ISREW	C4 Intelligence Surveillance Reconnaissance and Electronic Warfare
CCEB	Combined Communications Electronics Board (Australia, Canada, New Zealand, United Kingdom and NATO)
CSS	Cascading Style Sheet
DBA	Database Administrator
DSTO	Defence Science and Technology Organisation
GDN	Oracle Global Database Name
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transfer Protocol
JDBC	Java Database Connectivity
JDK	Java Development Kit
JWID	Joint Warrior Interoperability Demonstration
JRE	Java Runtime Environment
JSDK	Java Servlet Development Kit
JSP	Java Server Pages
SID	Oracle Service Identification
URL	Universal Resource Locator
WDCT	Web Data Collection Tool
XML	eXtensible Markup Language

# 1. Introduction

This document describes what the Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) is, and how to install and run it correctly. The WDCT is described in detail on how it has been built and the functionality of each component. In the installation section there is a description of how to install each part of the WDCT including the web server, servlet engine, Oracle database, and WDCT servlet files.

## 1.1. JWID Description

JWID is a multi national activity, which seeks to develop joint and combined C4 Intelligence Surveillance and Reconnaissance (C4ISR) warfighting capability and to improve system interoperability between participating nations. JWID runs in a two-year cycle with the first year having a “demonstration” theme, where the purpose is for relevant defence and industry participants to demonstrate the latest developments in technology. The second year is referred to as the “exploitation” period where the focus is on exploiting from a military perspective, the best performing demonstrations as identified through the assessment process from the first year of the cycle. The “exploitation” process is based around “fast tracking” the inclusion of those selected technologies into active service. As can be seen, the assessment process plays a pivotal role in the JWID process.

JWID is a C4ISR technology demonstration and exploitation activity within a simulated operational setting, where the focus is getting warfighter feedback on demonstration activities’ operational utility, effectiveness and relevance as an Australian Defence Force (ADF) C4 Intelligence Surveillance Reconnaissance and Electronic Warfare (C4ISREW) capability rather than winning the battle. The primary aim of JWID is to provide a risk free experimentation environment to expose and develop solutions for joint and combined warfighting problems. JWID 2000’s primary focus was to demonstrate off-the-shelf, new and evolving capabilities that may solve C4ISR deficiencies and interoperability issues for the Australian warfighter. JWID 2000 operated within a coalition environment comprising real, virtual and simulated strategic, operational and tactical forces from the CCEB (Combined Communications Electronics Board (Australia, Canada, New Zealand, United Kingdom and NATO)) nations against a determined virtual aggressor.

During a JWID activity, warfighters and technologists interact to develop and assess new capabilities, to inform Australia’s C4ISR capability development process. At the same time, theatre warfighting personnel assessed each capability and provided valuable feedback to Demonstration sponsors on the warfighting effectiveness, operational utility and relevance of the capability being demonstrated.

## **1.2. JWID Assessment Process Description**

The role of the Australian Assessment Program for JWID 2000 was for warfighting staff using the demonstration facilities, as well as the demonstration sponsors (those organisations that supplied the demonstration) to provide feedback on the performance of the demonstration activities. This feedback was achieved through the use of web-based technology to electronically collect warfighter's feedback based upon a set of pre-determined assessment criteria. These assessment criteria had been developed from a series of requirements established by the demonstration sponsor and the warfighting sponsor (Headquarters Australian Theatre (HQAST)). JWID 2000 was the first time an Australian assessment process had been formally conducted with the aim of encompassing the majority of demonstration activities. Prior to JWID 2000 individual demonstrations may have been assessed, however there had been no overarching assessment process as part of Australia's involvement in the JWID program.

It must be stressed that the assessment program proposed for JWID 2000 had been designed around an evaluation that would be based on both warfighter and engineering assessments and not by an independent assessment team or by the assessment manager. One of the formal aims for the assessment process was to establish a process and methodology that could be implemented for JWID 2000 (and the future). The intent of the process was to create a methodical approach to the development of a series of endorsed assessment criteria and a mechanism by which the warfighter and engineering assessments could be easily recorded for subsequent collation. The assessment team did not conduct the assessments but merely facilitated the development of the assessment criteria and the conduct of these assessments. For more information on the JWID assessment process please contact Paul Rogers of DSTO (Defence, Science & Technology Organisation, paul.rogers@dsto.defence.gov.au).



### 1.3. WDCT Description

The WDCT consists primarily of a Jakarta Tomcat servlet container operating in a standalone server mode and a database. Jakarta Tomcat 'serves' the web pages to the client (Microsoft Internet Explorer web browser) for a warfighter to view or input the information stored and retrieved from the database. Figure 1 below diagrammatically represents the operation of the WDCT.

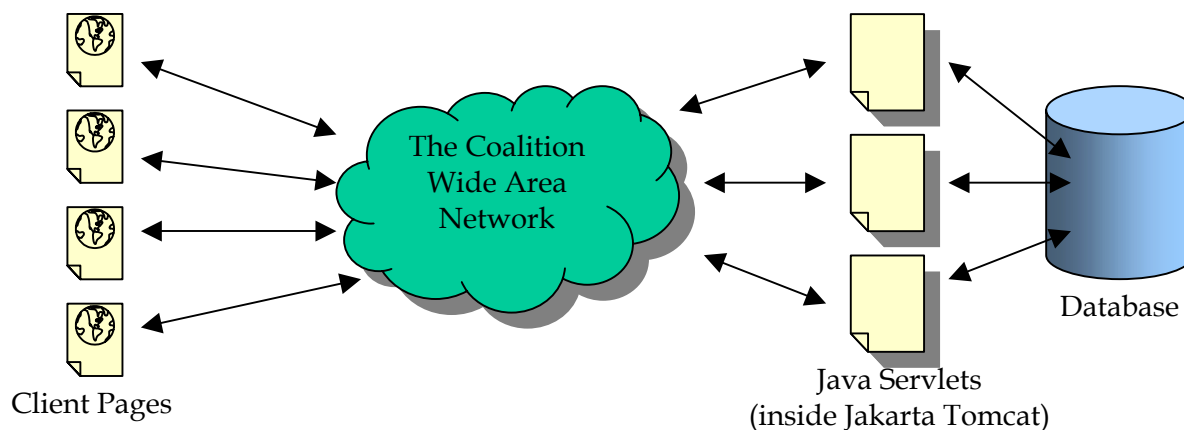


Figure 1 - Diagram of how the WDCT works

Jakarta Tomcat is freely available off the Internet from the Apache Software Foundation. To insert and retrieve information to and from the database Java Servlets are used. The Java programming language has been used to write the Java Servlets. Within the Java Language is the JDBC (Java Database Connectivity) data access API (Application Programming Interface). JDBC allows the servlets to connect to the database so information can be retrieved and stored. The servlets communicate between the client and the server by outputting Hyper Text Mark-up Language (HTML) and JavaScript over Hyper Text Transfer Protocol (HTTP) to the client browser. JavaScript scripts are used on the client for manipulation of visual information and data.

The database used for the WDCT is Oracle 8i. Oracle supports direct connections from JDBC. The Oracle database is used to store all the configuration and analysis information. Administration of the database and other WDCT functions is also done via the web interface.

A description of how to set up each component of the WDCT is described later in this document.

## 2. Installation

### 2.1. Installation Introduction

There are four components to the WDCT system; the Java platform, the Jakarta Tomcat, the Oracle Database and the WDCT files. Although as much detail on the installation procedures is given as possible some items may vary slightly depending on the operating system used and versions of products used.

In this manual there are numerous places where you must supply your own information. This is indicated by a name surrounded by square brackets. For example [CD-ROM Drive] means that you should replace the text “[CD-ROM Drive]” with the drive letter of your CD-ROM.

#### Java

Java is the main component of the WDCT. All the functionality is written as Java Servlets and JSP (Java Server Pages), therefore Java is required for WDCT functionality. JDK 1.3.1 (Java Development Kit) is the recommended version of the Java 2 platform. The JDK contains all the main Java development environment and runtime files.

#### Jakarta Tomcat

The web server used for the WDCT system is the Jakarta Tomcat 3.2.3 servlet container. This can be either downloaded from the WDCT CD-ROM or from the Internet. The web site for the Jakarta Tomcat is <http://jakarta.apache.org/tomcat/>. The web server is used to accept all requests from a client. Jakarta Tomcat is not only capable of serving static web content but also Java Servlet and JSP dynamic content within the WDCT system.

#### Oracle

Oracle is the database system used to store all analysis information collected through the WDCT web interface. Most configuration settings are also contained within the database. All question and demonstration details are also contained within the database. The Java Servlets talk to the database via JDBC.

#### WDCT Files

The WDCT files contain all the images, HTML (HyperText Markup Language), CSS (Cascading Style Sheets), JavaScript, JSP, XML (eXtensible Markup Language) and compiled Servlet Class files that are used within the WDCT server. All WDCT files are stored in a specific directory structure called a context that configures and defines how the files are accessible over the Web and restricts user access to files where necessary.

## 2.2. Installation Procedure

### 2.2.1. Java

The whole of the WDCT runs on the Java 2 platform using Java Servlets and JSP, therefore Java 2 must be installed. The version of Java 2 used is JDK (Java Development Kit) version 1.3.1. This can be found at:

On the web:

<http://Java.sun.com/>

On WDCT CD-ROM:

[CD-ROM Drive]:\Java\j2sdk-1\_3\_1-win.exe

Within the JDK are all the files to compile Java programs, the Java API specification, and the Java Runtime Environment (JRE). The JRE is used to run the Java programs.

#### 2.2.1.1. To install Java

1. Run the "j2sdk-1\_3\_1-win.exe" file that is on the WDCT CD-ROM or downloaded from the Internet. This will start the installation program for the JDK.
2. Click the "Next" button at the welcome screen.
3. Read the Sun software license agreement. Click the "Yes" button to accept the Sun license agreement.
4. Click "Next" to accept installing the JDK into the "c:\jdk1.3.1\" directory.
5. Click "Next" to install all parts of the JDK.
6. Java will now install onto your computer.
7. Click the "Finish" button to exit the Java install program and restart the computer.

Now the JDK and JRE are installed. The home for the JDK install is:

c:\jdk1.3.1\

The next step is to install and configure the Oracle database.

### 2.2.2. Oracle

This installation procedure was constructed using Oracle 8i.1.7 although Oracle 9i has also been used. There may be slight differences between different versions of Oracle, therefore it is recommended that an Oracle DBA (Database Administrator) be used to install, configure and maintain the Oracle database. Installation instructions are given below for both Oracle 8i.1.7 and Oracle 9i.

#### 2.2.2.1. Install Oracle 8i.1.7 Software

1. Insert the Oracle Enterprise Installation CD into your computer. The installer should start running automatically. If it does not, run the setup.exe file on the CD-ROM.
2. Click the "install/deinstall products" button to enter the install wizard.
3. Click "Next" to start installing.
4. Click "Next" to accept the default source and destination for the Oracle install. The name should be "OraHome81" and the path should be "C:\Oracle\Ora81\".
5. The product list should now load. This may take some time to complete.
6. Click "Next" button to install the "Oracle 8i Enterprise Edition" version as both the server software and the management software are needed to use and maintain the database.
7. Click the "Next" button to select the "typical (779MB)" type of installation. If you have hard drive space limitations the installation can be customised to remove certain parts of the installation. For more information on customising the install please contact Egon Kuster (egon.kuster@dsto.defence.gov.au).
8. Enter the Global Database Name (GDN) in the format "[database name].[domain name]" (for example "wdct.dsto.defence.gov.au").
9. The SID (Service Identification) is automatically completed, the SID and GDN (Global Database Name) are used to identify the database on the network. Do not forget to record the SID and GDN you have entered for future reference.
10. Click "Next" to show the summary of the installation options.
11. Click the "Install" button to install the software with the selected configuration. Oracle will now install. This could take quite some time to complete. The installation will also create a database instance with two new users, SYS and SYSTEM with the username/password of:  
     SYS/change\_on\_install  
     SYSTEM/manager  
 These users' passwords should be changed as soon as possible after the installation is complete.
12. Click "Exit" to complete the install and exit the install program.
13. Click "Yes" to confirm your exit. You now have a full copy of Oracle installed.

#### 2.2.2.2. Install Oracle 9i Software

1. Insert the Oracle 9i Server installation Disk 1 CD-ROM into the server. If the installer does not start automatically, run the "setup.exe" file on the CD-ROM.
2. Click the "Next" button to start the install process.
3. The next screen outlines the location of the source install files and the installation destination. Leave these fields as the default.  
     Source Path: [CD-ROM Drive]:\stage\products.jar  
     Destination Name: OraHome90  
     Destination Path: c:\oracle\ora90
4. Click the "Next" button to accept the source and destination paths.
5. A product list will now load; this could take quite some time to complete.

6. Three options are displayed. Select "Oracle 9i Database 9.0.1.1.1" and click the "Next" button.
7. Next you are asked what type of installation is needed, select "Enterprise Edition" and click the "Next" button. This version of the installation includes all the software needed to create and maintain the database.
8. The installer now asks how to configure the database. The WDCT is a transaction based system, therefore the "Transaction Processing" choice should be selected. After selecting this choice click the "Next" button.
9. To find the database on the network a type of identifier is needed. For Oracle this is the GDN and SID. The GDN is typically the database name followed by the domain the machine is within for example "wdct.dsto.defence.gov.au". Enter the GDN into the supplied field.
10. After entering the GDN the SID is automatically completed based on the first portion of the GDN. Accept the results by clicking the "Next" button.
11. Next the installer asks for the location to save the database files. Accept the default location by clicking the "Next" button.
12. Oracle 9i can support multiple character sets for storing and displaying text. Accept the default setting by clicking the "Next" button.
13. When the summary page is displayed, review the information to verify it is correct then click the "Install" button to start copying the files across to the server. This is a good time to go and have a coffee as it takes a long time to complete.
14. When prompted insert Disk 2 and then press the "Ok" button to continue.
15. When prompted insert Disk 3 and then press the "Ok" button to continue.
16. Record the database information as displayed for future reference, then click the "Exit" button to complete the installation.
17. Click the "Exit" button to quit the Oracle installer and complete the installation. You now have a full copy of Oracle installed.

#### 2.2.2.3. Disable the Oracle HTTP Server

Depending on the version of Oracle you are installing, an Oracle HTTP server might have been configured to run on your system. Because this will make use of the standard HTTP port, the WDCT system will not function correctly. Therefore the Oracle HTTP server must be disabled. To check and disable the server if necessary:

1. Open the Control Panel by clicking Start->Settings->Control Panel.
2. In the newly opened Control Panel window, locate and double click the "Administrative Tools" icon.
3. In the newly opened Administrative Tools window locate and double click the "Services" icon.
4. In the services window, locate the "Oracle[Oracle Home]HTTPServer". The Oracle home name is by default "OraHome81" or "OraHome90".
5. If this is not present you can skip the remaining steps and proceed with Configuring Oracle section.

6. If a service "Oracle[Oracle Home]HTTPServer" is present right click on the service and select "Stop" from the context. If the service is not running already proceed to the next step without action.
7. Right click on the "Oracle[Oracle Home]HTTPServer" service and select "Properties" from the context menu.
8. Find the "Startup type" drop down selection box and ensure that the startup type is "Manual" or "Disabled".
9. Click "OK" to close the "Oracle[Oracle Home]HTTPServer" Properties window.
10. Close the Services and Administrative Tools windows by clicking the "X" button located in the top right hand corner of each window.

Oracle is now installed. Oracle still needs to be configured for use with the WDCT.

#### 2.2.2.4. Configuring Oracle

All configuration is done by logging into an Oracle administration tool of choice as the SYS user. For example Oracle Enterprise Manager or DBA Studio could be used. For more information on conducting these configuration options check the Oracle user guide.

1. Change the passwords for the SYS and SYSTEM users, record the new passwords in a safe place.
2. Delete or disable the SCOTT user account. SCOTT is the default user for Oracle and is created when Oracle is installed.
3. Create a new tablespace with the name "WDCT".

The sample SQL for creating a new tablespace:

```
CREATE TABLESPACE wdct DATAFILE
'c:\oracle\ORADATA\wdct\wdct.dbf' SIZE 50M AUTOEXTEND ON MAXSIZE
UNLIMITED ONLINE;
```

4. Create the WDCT user account. This user account is used to allow the Java Servlets to connect to the database, edit, alter and update information. Create the user account with the following attributes:

```
Username:      WDCTUSER
Roles:         CONNECT
System Priveledges: EXECUTE ANY PROCEDURE
                INSERT ANY TABLE
                SELECT ANY SEQUENCE
                SELECT ANY TABLE
                UPDATE ANY TABLE
Tablespace:    WDCT
Quota:         Unlimited on WDCT, TEMP
Object Priveledges: None
```

The sample SQL for creating the new user is:

```
CREATE USER "WDCTUSER" PROFILE "DEFAULT" IDENTIFIED BY
"pass00word" DEFAULT TABLESPACE "WDCT" TEMPORARY TABLESPACE
"TEMP" QUOTA UNLIMITED ON WDCT ACCOUNT UNLOCK QUOTA UNLIMITED ON
TEMP ACCOUNT UNLOCK;
```

```
GRANT EXECUTE ANY PROCEDURE TO "WDCTUSER";
GRANT INSERT ANY TABLE;
GRANT SELECT ANY SEQUENCE TO "WDCTUSER";
GRANT SELECT ANY TABLE TO "WDCTUSER";
GRANT UPDATE ANY TABLE TO "WDCTUSER";
GRANT "CONNECT" TO "WDCTUSER";
```

Do not forget to record the password used to create the new user account. Close the administration tool. Next the default tables and values need to be entered into the database.

#### 2.2.2.5. Create WDCT Tables and Populate Initial Values

Before using the database the default tables and initial values must be inserted into the database. This is done by following this procedure:

1. Start a command prompt window (Start -> Run, enter "cmd" then click "Run");
2. Change to "[CD-DRIVE]:\oracle", within this directory is the file "create\_db.sql". This file contains a number of SQL statements that when executed will create all the tables needed and populate the initial values in the tables.
3. Run the command "sqlplus" from the command line.
4. SQL Plus will now start up, login as the WDCTUSER user with the password that you assigned for WDCTUSER earlier (this document specifies this as "pass00word". SQL Plus is used to enter SQL statements directly to the database. It can also use SQL files that contain a number of SQL statements and run them like a batch file.
5. Now that you have logged into SQL Plus you can run the "create\_db.sql" file. Type "@create\_db" from the command prompt. This will load the file and run all the SQL statements contained within. If you started SQL Plus within the directory that contained the create\_db.sql file then it will run perfectly. If you ran SQL Plus from a different directory then the full path "@[CD-DRIVE]:/oracle/create\_db" is needed.
6. When the "create\_db.sql" runs a number of logs will be displayed. Ignore any error messages regarding "error dropping table|sequence". These errors occur because within the "create\_db.sql" file are statements to drop the tables and sequences in the database, if these do not exist then errors occur. To make sure that the "create\_db.sql" file worked, you can run it a second time and no errors should be displayed. If errors still exist then either the user that you are currently logged in as does not have sufficient privileges or there is an error in the database's configuration settings. After checking these two items and you are still having errors then contact Egon Kuster (egon.kuster@dsto.defence.gov.au).
7. Type "Exit" and hit enter to exit SQLPlus.

The database is now completely configured. The next step is to install and configure Jakarta Tomcat.

### 2.2.3. Jakarta Tomcat

To install Jakarta Tomcat you must have Winzip installed on your system. To find out if you have Winzip installed click the right mouse button on a file on your hard drive and if there is an option to “add to zip” in the context menu then you have Winzip installed. If you do not have the “add to zip” option displayed then you must first install Winzip before continuing. Winzip versions 7 or 8 can be installed. However, only version 8 is contained on the WDCT CD-ROM.

#### 2.2.3.1. Install WinZip

Winzip can be found at:

On the Internet:

<http://www.winzip.com/>

On the WDCT CD-ROM:

[CD-ROM Drive]:\winzip\WinZip80.exe

To install Winzip 8:

1. Run the winzip80.exe file that is on the WDCT CD-ROM or downloaded off the Internet.
2. Click the “Setup” button in the Winzip 8.0 setup window.
3. Click “OK” to select the “C:\Program Files\WinZip” path for installation.
4. Click the “Next” button to continue installation having read the feature list.
5. Click the “Yes” button to agree to the License Agreement and Warranty Disclaimer.
6. Click the “Next” button to continue. You may want to print the Quick Start Guide at this screen if you are not familiar with the WinZip software.
7. Select the “Start with WinZip Wizard” radio button. Then click the “Next” button.
8. Select the “Quick Search (faster)” radio button. Then click the “Next” button.
9. Click the “Next” button to list zip files in the “Favourite Zip Folders”.
10. Click the “Next” button to associate WinZip with supported file types.
11. Click the “Close” button to complete the installation.

Winzip is now installed.

To Install Winzip 7:

1. Run the winZip70.exe file that can be downloaded off the Internet.
2. This file is a self-extracting file and therefore asks for a temporary directory to extract into. The default temporary directory is “c:\temp”.
3. Click the “Unzip” button to extract all the files into the temporary directory.
4. Click the “OK” button to confirm that the files have been extracted.
5. Click the “Close” button to close the self-extracting program.
6. Run the “Setup.exe” file that has been extracted into the “c:\temp” directory.
7. Click the “OK” button to install Winzip into “c:\program files\winzip”.



8. Click the "Next" button to start the Winzip installation wizard.
  9. Click the "Yes" button to accept the Winzip license agreement.
  10. Select the "Start with WinZip Classic" option and then click the "Next" button.
  11. Click the "Next" button for "Express Setup".
  12. Click the "Finish" button to complete the installation and exit.
- Winzip is now installed.

#### 2.2.3.2. *Install Jakarta Tomcat*

All the software for the Jakarta Tomcat can be downloaded off the Internet. All downloads are also contained on the WDCT CD-ROM.

##### Web Sites:

<http://jakarta.apache.org/tomcat/>

##### On WDCT CD-ROM:

[CD-ROM Drive]:\Jakarta\jakarta-tomcat-3.2.3.exe

To install Jakarta Tomcat:

1. Locate the "jakarta-tomcat-3.2.3.zip" file on the WDCT CD-ROM or your download directory using Windows Explorer.
2. Right click on the "jakarta-tomcat-3.2.3.zip" file and select the "Extract to..." option from the context menu.
3. Click the "I Agree" button at the WinZip evaluation prompt if it appears.
4. Type "c:\\" or a different drive letter to extract the Tomcat software into. Make sure the "All files" radio button is selected and the "Use folder names" check box is checked.
5. Press the "Extract" button to begin extraction of the Jakarta Tomcat software.
6. Close the WinZip window by clicking the "X" button in the top right hand corner of its window.
7. A "jakarta-tomcat-3.2.3" directory has been created on the drive of your choice (C drive is the default for this installation procedure). This directory contains all the files required to run the Jakarta Tomcat software.

The installation of the Jakarta Tomcat core files is now complete. The environment variables that will be used by the Jakarta Tomcat must now be set up.

#### 2.2.3.3. *Set up Jakarta Tomcat Environment Variables*

Before using Jakarta Tomcat a number of environment variables must be set so that the software will work correctly. As Microsoft Windows 2000 and Microsoft Windows NT use different methods of setting the environment variables, both instructions are included below.

For Windows 2000:

1. Launch the control panel by clicking Start->Settings->Control Panel. This will bring up the control panel for the system.
2. Double click on the "System" icon in the control panel.
3. Select the "Advanced" tab from the tabs available at the top of the System Properties window.
4. Click the "Environment Variables" button to configure the system environment variables.
5. Click the "New..." button located below the System Variables section.
6. In the "New System Variable" window that appears enter "JAVA\_HOME" in the Variable Name text entry box and the root directory of your JDK installation ("C:\jdk1.3.1" is the default for this installation procedure) in the Variable Value text entry box. Click "OK".
7. Click the "New..." button in the Environment Variables located below the System Variables section.
8. In the "New System Variable" window that appears enter "TOMCAT\_HOME" in the Variable Name text entry box. In the Variable Value text entry box enter the root directory of your Jakarta Tomcat ("C:\jakarta-tomcat-3.2.3" is the default for this installation procedure). Click "OK".
9. You should now be able to see the "JAVA\_HOME" and "TOMCAT\_HOME" entries in the system variables scroll area, you may need to scroll up or down.
10. Exit the "Environment Variables" window by clicking the "OK" button.
11. Exit the "System Properties" window by clicking the "OK" button.

For Windows NT:

1. Launch the control panel by clicking Start->Settings->Control Panel. This will bring up the control panel for the system.
2. Double click on the "System" icon in the control panel.
3. Select the "Environment" tab from the tabs available at the top of the System Properties window.
4. Select an entry within the System variables list at the top of the dialog box. This is to give focus to the System variables box so that the next new variable will become a system variable.
5. In the Variable text field at the bottom of the window enter the text "JAVA\_HOME".
6. In the Value field enter the location of the root directory of your JDK installation directory (the default is "C:\jdk1.3.1").
7. Click the "Set" button to insert the new "JAVA\_HOME" system environment variable.
8. In the Variable text field now enter the text "TOMCAT\_HOME".
9. In the Value text field enter the directory where Tomcat was installed (the default location is "C:\jakarta-tomcat-3.2.3").
10. Click the "Set" button to insert the new "TOMCAT\_HOME" system environment variable.
11. You should now be able to see the "JAVA\_HOME" and "TOMCAT\_HOME" system environment variables within the system scroll list.
12. Exit the "Environment Variables" window by clicking the "OK" button.

13. Exit the "System Properties" window by clicking the "OK" button.

The Jakarta Tomcat environment variables are now set up. The next step in the installation procedure is to install the WDCT files.

#### 2.2.4. WDCT Files

To install all the WDCT files you need to create a context within Tomcat. The WDCT files are distributed as a ZIP file that contains a file structure needed to define a context. The default installation deploys the WDCT system so that typing "http://localhost" from the installation machine's browser and "http://[host].[domain]" from a remote machine's browsers will access the WDCT.

To perform this deployment:

1. Locate the [TOMCAT\_ROOT]\webapps directory using Windows Explorer or My Computer icon on your desktop.
2. In this directory create a ROOT directory. To do this rightclick inside the [TOMCAT\_ROOT]\webapps directory and select New->Folder from the context menu.
3. With the name of the newly created directory still highlighted, type ROOT, using all capital letters and press enter to set the directory name.
4. Locate the WDCT-3.0.zip file contained on the WDCT CD-ROM in the "wdct" directory. Right click on this file and select the "Extract to " option from the context menu.
5. Click "I Agree" if the WinZip evaluation notice is displayed.
6. In the "Extract To..." text entry box type "[TOMCAT\_HOME]\webapps\ROOT\". Alternatively you may use the select "Folders/Drives:" window to select the ROOT directory inside your Jakarta Tomcat webapps directory.
7. Make sure that the "All Files" radio button is selected and press the "Extract" button to extract the files and directories that make up the context.
8. You may now close the WinZip window by clicking the "X" button located in the upper right hand corner of its window.

The WDCT context is now fully deployed as the root context for Jakarta Tomcat. This will allow the WDCT system to be accessed by using the "http://[host].[domain]" URL. The next step is to configure the installed components.

#### 2.2.5. Configure Components

The next step in setting up the system is to configure the Jakarta Tomcat as a standalone web server and make use of the database connection. The "server.xml"

file contains all relevant Jakarta Tomcat settings and is found in "[TOMCAT\_HOME]\conf" directory. The "web.xml" file contains all the database settings relevant to the context and is found in "[TOMCAT\_HOME]\webapps\ROOT\WEB-INF\" directory. A reference copy of these files with the default changes applied can be found in the appendixes.

#### 2.2.5.1. Edit Configuration Files

The following changes need to be applied to the "[TOMCAT\_HOME]\conf\server.xml" file:

1. Following the text:

```
<RequestInterceptor
  className="org.apache.tomcat.request.InvokerInterceptor"
  debug="0" prefix="/servlet/" />
```

this text needs to be added:

```
<RequestInterceptor
  className="org.apache.tomcat.request.InvokerInterceptor"
  debug="0" prefix="/servlets/" />
```

2. In the following block:

```
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
  <Parameter name="handler"
    value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
  <Parameter name="port"
    value="8080"/>
</Connector>
```

The line containing value="8080" needs to be changed to value="80". This will configure the server to use the standard HTTP port.

The following changes may need to be applied to the

[TOMCAT\_HOME]\webapps\ROOT\web.xml file:

1. In the text :

```
<param-value>jdbc:oracle:thin:@localhost:1521:WDCT</param-value>
```

- The localhost needs to be replaced with the host name of the machine containing Oracle, for example "wdct.dsto.defence.gov.au".
- The "WDCT" part of the text needs to be the same as the SID that you have given to the database when during the Oracle installation. "WDCT" is the default SID for this installation manual.

2. In the text:

```
<param-name>UserName</param-name>
<param-value>wdctuser</param-value>
```

- wdctuser should be the database user created in section 2.2.2.4 Configuring Oracle, wdctuser is the default user described in this installation manual.

3. In the text:

```
<param-name>Password</param-name>
<param-value>pass00word</param-value>
```

- "pass00word" should be the password you assigned in section 2.2.2.4 Configuring Oracle; "pass00word" is the default for this installation procedure.

### 2.2.5.2. Oracle JDBC drivers for Jakarta Tomcat

The Oracle JDBC driver files must be copied into the Jakarta Tomcat context lib directory for Tomcat to make use of them.

1. Locate the file "classes12.zip" found in "[ORACLE\_HOME]\jdbc\lib" ("c:\oracle\ora81" or "c:\oracle\ora90" are the default oracle home directories for this installation manual) using Windows Explorer or the My Computer icon on your desktop.
2. Rightclick on the classes12.zip file and select "Copy" from the context menu.
3. Locate the Jakarta Tomcat ROOT context, lib directory. This is at "[TOMCAT\_ROOT]\Webapps\ROOT\WEB-INF\lib" ("c:\jakarta-tomcat-3.2.3\" is the default Tomcat root directory for this installation procedure). The file can't be placed into the lib directory within the tomcat root, it must be within the contexts' lib directory. If this file is not placed in the correct directory then errors will occur when starting Tomcat as it will not be able to locate the Oracle drivers.
4. Right click on the lib directory and select "Paste" from the context menu. This will copy the "classes12.zip" file into the lib directory.
5. Right click on the "classes12.zip" file you have just copied into the lib directory. Select "Rename" from the context menu.
6. Change the filename of the "classes12.zip" file to "classes12.jar".
7. Click "Yes" on the Rename prompt box if it appears.

### 2.2.5.3. Set Up Jakarta Tomcat To Run As A Windows Service

When you start the machine to run the WDCT you want to have Jakarta Tomcat automatically start running and stay running while the server is powered up. In Microsoft Windows NT you can accomplish this by registering Jakarta Tomcat as a service. To install Jakarta Tomcat as a service you need to:

1. Locate the "[CD-DRIVE]:\Jakarta\tomcat\_as\_a\_service.zip" file using Windows Explorer or from the "My Computer" icon.
2. Rightclick on this file and select "Extract to" option from the context menu.
3. Click "I Agree" if the WinZip evaluation notice is displayed.
4. In the "Extract To..." text entry box type "[TOMCAT\_HOME]\bin". Alternatively you may use the select "Folders/Drives:" window to select the bin directory inside your Jakarta Tomcat installation.
5. Make sure that the "All Files" radio button is selected and press the "Extract" button to extract the files. You should now have TomcatService.exe and Tomcat321service.bat in your "[TOMCAT\_HOME]\bin" directory.
6. You may now close the WinZip window by clicking the "X" button located in the upper right hand corner of its window.
7. Locate the "Tomcat321service.bat" file in "[TOMCAT\_HOME]\bin" directory using Windows Explorer or the My Computer icon on your desktop.

8. Double click on the "Tomcat321service.bat" file. This will install Jakarta Tomcat as a service on your machine. The service name will be Jakarta. The service will be configured to start automatically.
9. Open the Control Panel by clicking Start->Settings->Control Panel.
10. Locate the "Administrative Tools" icon within the newly opened Control Panel window.
11. Double click the "Administrative Tools" icon to open the Administrative Tools window.
12. Locate the "Services" icon within the Administrative Tools window and double click on this icon to open the Services window. This will allow you to view the status of all the services on our system.
13. Locate the "Jakarta" service. Its startup type should be automatic. If the status is blank, or does not read "Started", click on the start button while the service is highlighted to start it.
14. The Jakarta Tomcat should now be fully functional as a service.

#### **2.2.6. Microsoft Internet Explorer 5.5**

The recommended browser for accessing the WDCT system is Internet Explorer 5.5 . This Web browser will ensure that all the WDCT functionality is correct. This is the browser which was used during development and testing of the WDCT system. This browser needs to be installed on all the client machines used to access the WDCT system. To install this version of Internet Explorer:

1. Locate the "[CD-ROM Drive]:\IE5.5\" directory on the WDCT CD-ROM using Windows Explorer or the My Computer icon on your desktop.
2. Make yourself familiar with the "IE.TXT" and "README.TXT" files in that directory. To open these files simply double click on them.
3. Double click on the "i386" directory. In this directory locate the "IE5SETUP.EXE" application file.
4. Double click on the "IE5SETUP.EXE" to begin installing Internet Explorer 5.5 on your system.
5. Read the End User License Agreement and select the "I accept the agreement" radio button. Click the "Next" button to continue the installation.
6. Click "Next" to continue the installation.
7. Click the "Finish" button to complete the installation of Internet Explorer 5.5 and restart the computer.

You have now completed the installation of the WDCT system on your machine. To access the system, enter "http://[hostname].[domain]/" into the Microsoft Internet Explorer browser.

### 3. Multiple Instances

All the instructions so far have been on how to set up a single instance of the WDCT running on one machine with one database. It is also possible to run two or more versions of the WDCT with one database. Below are instructions on converting from one instance to multiple instances. Multiple instances must be created within Jakarta Tomcat and the database altered to accommodate the new instances. It is also possible to have each instance access separate databases. Although this is not described, it only needs the change of the connection string within the web.xml file to the new databases.

#### 3.1. Jakarta Tomcat

Currently the WDCT is installed into the ROOT context of the Tomcat server. This will need to be changed so that there are separate contexts for each instance of the WDCT. Within Tomcat the ROOT context is special as it gets loaded when the server is accessed by a browser without any directory information at the end of the URL. For instance "http://hostname.domain/" will access the "ROOT" context, while "http://hostname.domain/jwid/" will access the "jwid" context.

Installing multiple WDCT instances:

1. Stop the Jakarta service by typing "net stop jakarta" at the command prompt.
2. Rename the ROOT directory in the Tomcat webapps directory to the name of the first instance (for example "jwid").
3. Locate the WDCT-3.0.zip file contained on the WDCT CD-ROM in the "wdct" directory. Right click on this file and select the "Extract to " option from the context menu.
4. Click "I Agree" if the WinZip evaluation notice is displayed.
5. In the "Extract To..." text entry box type "[TOMCAT\_HOME]\webapps\[New Context]\". This will extract a new version of the WDCT software into a new context. Repeat steps 4 and 5 for all the new instances of the WDCT that you wish to create.
6. For each new instance created edit its "web.xml" file to update the database connection string, username, and password. Each instance will need its own username and password to connect to the database. Keep a record of the name of each context, the username and password so that it can be used when editing the database.
7. The ROOT directory within the webapps directory needs to be recreated and within this a splash page should be developed. This splash page should contain links to all the other WDCT contexts. The splash page should be saved as "index.html" so that it will be automatically loaded when the machine's URL is entered into the browser.

8. Before restarting the Tomcat server the database must also be altered to handle multiple instances (as shown in section 3.2 on page 18). (To restart Tomcat type "net start jakarta" at the command prompt)

### 3.2. Oracle

Along with creating new contexts for each new instance of the WDCT within the Tomcat service there is also a need to edit the database. For each new context a tablespace and user must be created. The tablespace will contain all the data files for each WDCT instance and each user is linked to a tablespace.

Alter Oracle database:

1. Start your favourite Oracle administration tool (for example Oracle Enterprise Manager or DBA Studio).
2. For each new instance create a tablespace with the same name as the context.

The sample SQL for creating a new tablespace is:

```
CREATE TABLESPACE wdct DATAFILE
'c:\oracle\ORADATA\wdct\wdct.dbf' SIZE 50M AUTOEXTEND ON MAXSIZE
UNLIMITED ONLINE;
```

3. Create all the user accounts, one for each new context as defined within the "web.xml" files. Create the user accounts with the following attributes:

```
Username:      [Context Name]USER
Roles:         CONNECT
System Priveledges: EXECUTE ANY PROCEDURE
                INSERT ANY TABLE
                SELECT ANY SEQUENCE
                SELECT ANY TABLE
                UPDATE ANY TABLE
Tablespace:    [Context's Tablespace]
Quota:         Unlimited on [Context's Tablespace], TEMP
Object Priveledges: None
```

The sample SQL for creating the new user is:

```
CREATE USER "WDCTUSER" PROFILE "DEFAULT" IDENTIFIED BY
"pass00word" DEFAULT TABLESPACE "WDCT" TEMPORARY TABLESPACE
"TEMP" QUOTA UNLIMITED ON WDCT ACCOUNT UNLOCK QUOTA UNLIMITED ON
TEMP ACCOUNT UNLOCK;
GRANT EXECUTE ANY PROCEDURE TO "WDCTUSER";
GRANT INSERT ANY TABLE;
GRANT SELECT ANY SEQUENCE TO "WDCTUSER";
GRANT SELECT ANY TABLE TO "WDCTUSER";
GRANT UPDATE ANY TABLE TO "WDCTUSER";
GRANT "CONNECT" TO "WDCTUSER";
```

4. Now you should have a new user and tablespace within the database for each instance of the WDCT installed.
5. The default tables and values will need to be entered into each new tablespace. To install the tables and default values, run the "create\_db.sql" script as before but



login using each new user and run the script once for each user. This will create the default tables within each tablespace.

6. Oracle is now configured to handle multiple instances. If Tomcat has been altered with the new instances, it can then be restarted and should allow access to all new WDCT instances.

### 3.3. Settings

To give a better understanding of the configuration needed for creating multiple instances of the WDCT this section will outline the settings needed for the creation of three WDCT instances named “jwid”, “cinc21” and “ctl”.

<b>Instance Name</b>	<b>jwid</b>
<b>Connection String</b>	jdbc:oracle:thin:@[hostname].[domain]:1521:WDCT
<b>Context Name</b>	jwid
<b>Context location</b>	[TOMCAT_HOME]\webapps\jwid
<b>Database Username</b>	jwiduser
<b>Database Tablespace</b>	jwid

<b>Instance Name</b>	<b>cinc21</b>
<b>Connection String</b>	jdbc:oracle:thin:@[hostname].[domain]:1521:WDCT
<b>Context Name</b>	cinc21
<b>Context location</b>	[TOMCAT_HOME]\webapps\cinc21
<b>Database Username</b>	cinc21user
<b>Database Tablespace</b>	cinc21

<b>Instance Name</b>	<b>ctl</b>
<b>Connection String</b>	jdbc:oracle:thin:@[hostname].[domain]:1521:WDCT
<b>Context Name</b>	ctl
<b>Context location</b>	[TOMCAT_HOME]\webapps\ctl
<b>Database Username</b>	ctluser
<b>Database Tablespace</b>	ctl

## 4. Troubleshooting

If the Tomcat service fails to start up:

1. Ensure that there are no other services configured to use port 80. If such services exist they will need to be disabled or reconfigured to make use of a different port. Alternatively the WDCT system can be configured to use a port other than 80. However this port will need to be explicitly specified in the URL string used to access the WDCT system.
2. Check that the Oracle database drivers are placed in the “lib” directory of the deployed context.
3. Check the Tomcat logs found in the “[TOMCAT\_HOME]\logs\” directory.

If the WDCT system behaves incorrectly this is usually manifested by the Contact Information to displaying correctly at the bottom of the WDCT pages:

1. Ensure that the Oracle username and password are correctly configured in the "web.xml" file.
2. Ensure that the "server.xml" contains the text:  
`<RequestInterceptor`

```
className="org.apache.tomcat.request.InvokerInterceptor"  
debug="0" prefix="/servlets/" />  
and that the XML structure is not malformed.
```

If you cannot log into the administrator area, this can be because your username and password are incorrect (within the "web.xml" file) or the WDCT system cannot connect to the Oracle database.

1. Check if the database is up and online by connecting to the database via SQLPlus on the command line.
2. Check the connection string, username and password within the "web.xml" file in the "ROOT" Tomcat context.

## Appendix A: server.xml listing

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<Server>
  <!-- Debug low-level events in XmlMapper startup
  <xmlmapper:debug level="0" />
  -->

  <!--

Logging:

    Logging in Tomcat is quite flexible; we can either have a log
    file per module (example: ContextManager) or we can have one
    for Servlets and one for Jasper, or we can just have one
    tomcat.log for both Servlet and Jasper. Right now there are
    three standard log streams, "tc_log", "servlet_log", and
    "JASPER_LOG".

Path:

The file to which to output this log, relative to
TOMCAT_HOME. If you omit a "path" value, then stderr or
stdout will be used.

Verbosity:

Threshold for which types of messages are displayed in the
log. Levels are inclusive; that is, "WARNING" level displays
any log message marked as warning, error, or fatal. Default
level is WARNING.

verbosityLevel values can be:
    FATAL
    ERROR
    WARNING
    INFORMATION
    DEBUG

Timestamps:

By default, logs print a timestamp in the form "yyyy-MM-dd
hh:mm:ss" in front of each message. To disable timestamps
completely, set 'timestamp="no"'. To use the raw
msec-since-epoch, which is more efficient, set
'timestampFormat="msec"'. If you want a custom format, you
can use 'timestampFormat="hh:mm:ss"' following the syntax of
Java.text.SimpleDateFormat (see Javadoc API). For a
production environment, we recommend turning timestamps off,
or setting the format to "msec".

Custom Output:

"Custom" means "normal looking". "Non-custom" means
"surrounded with funny xml tags". In preparation for
possibly disposing of "custom" altogether, now the default is
'custom="yes"' (i.e. no tags)

Per-component Debugging:

Some components accept a "debug" attribute. This further
enhances log output. If you set the "debug" level for a
component, it may output extra debugging information.
-->
```

```

<!-- if you don't want messages on screen, add the attribute
      path="logs/tomcat.log"
to the Logger element below
-->
<Logger name="tc_log"
        verbosityLevel = "INFORMATION"
/>

<Logger name="servlet_log"
        path="logs/servlet.log"
/>

<Logger name="JASPER_LOG"
        path="logs/jasper.log"
        verbosityLevel = "INFORMATION" />

<!-- You can add a "home" attribute to represent the "base" for
all relative paths. If none is set, the TOMCAT_HOME property
will be used, and if not set "." will be used.
webapps/, work/ and logs/ will be relative to this ( unless
set explicitelty to absolute paths ).

You can also specify a "randomClass" attribute, which determines
a subclass of Java.util.Random will be used for generating session IDs.
By default this is "Java.security.SecureRandom".
Specifying "Java.util.Random" will speed up Tomcat startup,
but it will cause sessions to be less secure.

You can specify the "showDebugEnabled" attribute to control whether
debugging information is displayed in Tomcat's default responses.
This debugging information includes:
    1. Stack traces for exceptions
    2. Request URI's that cause status codes >= 400
The default is "true", so you must specify "false" to prevent
the debug information from appearing. Since the debugging
information reveals internal details about what Tomcat is serving,
set showDebugEnabled="false" if you wish increased security.
-->
<ContextManager debug="0" workDir="work" showDebugEnabled="true" >

<!-- ===== Interceptors ===== -->

<!--
ContextInterceptor className="org.apache.tomcat.context.LogEvents"
-->

<ContextInterceptor className="org.apache.tomcat.context.AutoSetup" />

<ContextInterceptor
    className="org.apache.tomcat.context.WebXmlReader" />

<!-- Uncomment out if you have JDK1.2 and want to use policy
<ContextInterceptor
    className="org.apache.tomcat.context.PolicyInterceptor" />
-->

<ContextInterceptor
    className="org.apache.tomcat.context.LoaderInterceptor" />
<ContextInterceptor
    className="org.apache.tomcat.context.DefaultCMSetter" />
<ContextInterceptor
    className="org.apache.tomcat.context.WorkDirInterceptor" />

<!-- Request processing -->
<!-- Session interceptor will extract the session id from cookies and
deal with URL rewriting ( by fixing the URL ). If you wish to
suppress the use of cookies for session identifiers, change the
"noCookies" attribute to "true"
-->

```

```

<RequestInterceptor
  className="org.apache.tomcat.request.SessionInterceptor"
  noCookies="false" />

<!-- Find the container ( context and prefix/extension map )
    for a request.
-->
<RequestInterceptor
  className="org.apache.tomcat.request.SimpleMapper1"
  debug="0" />

<!-- Non-standard invoker, for backward compat. ( /servlet/* )
    You can modify the prefix that is matched by adjusting the
    "prefix" parameter below. Be sure your modified pattern
    starts and ends with a slash.

    NOTE: This prefix applies to *all* web applications that
    are running in this instance of Tomcat.
-->
<RequestInterceptor
  className="org.apache.tomcat.request.InvokerInterceptor"
  debug="0" prefix="/servlet/" />

<RequestInterceptor
  className="org.apache.tomcat.request.InvokerInterceptor"
  debug="0" prefix="/servlets/" />

<!-- "default" handler - static files and dirs. Set the
    "suppress" property to "true" to suppress directory listings
    when no welcome file is present.

    NOTE: This setting applies to *all* web applications that
    are running in this instance of Tomcat.
-->
<RequestInterceptor
  className="org.apache.tomcat.request.StaticInterceptor"
  debug="0" suppress="false" />

<!-- Plug a session manager. You can plug in more advanced session
    modules.
-->
<RequestInterceptor
  className="org.apache.tomcat.session.StandardSessionInterceptor" />

<!-- Check if the request requires an authenticated role.
-->
<RequestInterceptor
  className="org.apache.tomcat.request.AccessInterceptor"
  debug="0" />

<!-- Check permissions using the simple xml file. You can
    plug more advanced authentication modules.
-->
<RequestInterceptor
  className="org.apache.tomcat.request.SimpleRealm"
  debug="0" />

<!-- Uncomment the following and comment out the
    above to get a JDBC realm.
    Other options for driverName:
    driverName="oracle.jdbc.driver.OracleDriver"
    connectionURL="jdbc:oracle:thin:@ntserver:1521:ORCL"
    connectionName="scott"
    connectionPassword="tiger"

    driverName="org.gjt.mm.mysql.Driver"
    connectionURL="jdbc:mysql://localhost/authority"
    connectionName="test"
    connectionPassword="test"

```

```

        "connectionName" and "connectionPassword" are optional.
-->
<!--
<RequestInterceptor
    className="org.apache.tomcat.request.JDBCRealm"
    debug="99"
driverName="sun.jdbc.odbc.JdbcOdbcDriver"
connectionURL="jdbc:odbc:TOMCAT"
userTable="users"
    userNameCol="user_name"
    userCredCol="user_pass"
userRoleTable="user_roles"
    roleNameCol="role_name" />
-->

<!-- Loaded last since JSP's that load-on-startup use request handling -->
<ContextInterceptor
    className="org.apache.tomcat.context.LoadOnStartupInterceptor" />

<!-- ===== Connectors ===== -->

<!-- Normal HTTP -->
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
    <Parameter name="handler"
        value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
    <Parameter name="port"
        value="80"/>
</Connector>

<!--
    Uncomment this for SSL support.
    You _need_ to set up a server certificate if you want this
    to work, and you need JSSE.
    1. Add JSSE jars to CLASSPATH
    2. Edit Java.home/jre/lib/security/Java.security
        Add:
        security.provider.2=com.sun.net.ssl.internal.ssl.Provider
    3. Do: keytool -genkey -alias tomcat -keyalg RSA
        RSA is essential to work with Netscape and IIS.
        Use "changeit" as password. ( or add keypass attribute )
        You don't need to sign the certificate.

    You can set parameter keystore and keypass if you want
    to change the default ( user.home/.keystore with changeit )
-->
<!--
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
    <Parameter name="handler"
        value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
    <Parameter name="port"
        value="8443"/>
    <Parameter name="socketFactory"
        value="org.apache.tomcat.net.SSLSocketFactory" />
</Connector>
-->

<!-- Apache AJP12 support. This is also used to shut down tomcat.
-->
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
    <Parameter name="handler"
value="org.apache.tomcat.service.connector.Ajp12ConnectionHandler"/>
    <Parameter name="port" value="8007"/>
</Connector>

<!-- ===== Special webapps ===== -->
<!-- You don't need this if you place your app in webapps/
and use defaults.
For security you'll also need to edit tomcat.policy

```

Defaults are: debug=0, reloadable=true, trusted=false  
 (trusted allows you to access tomcat internal objects  
 with FacadeManager ), crossContext=true (allows you to  
 access other contexts via ServletContext.getContext())

If security manager is enabled, you'll have read perms.  
 in the webapps dir and read/write in the workdir.

-->

```
<Context path="/examples"
        docBase="webapps/examples"
        crossContext="false"
        debug="0"
        reloadable="true" >
</Context>
```

<!-- Admin context will use tomcat.core to add/remove/get info about  
 the webapplications and tomcat internals.  
 By default it is not trusted - i.e. it is not allowed access to  
 tomcat internals, only informations that are available to all  
 servlets are visible.

If you change this to true, make sure you set a password.

-->

```
<Context path="/admin"
        docBase="webapps/admin"
        crossContext="true"
        debug="0"
        reloadable="true"
        trusted="false" >
</Context>
```

<!-- Virtual host example -  
 In "127.0.0.1" virtual host we'll reverse "/" and  
 "/examples"  
 (XXX need a better example )  
 (use "http://127.0.0.1/examples" )

```
<Host name="127.0.0.1" >
    <Context path=""
            docBase="webapps/examples" />
    <Context path="/examples"
            docBase="webapps/ROOT" />
</Host>
```

-->

```
</ContextManager>
</Server>
```

## Appendix B: web.xml listing

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://Java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>
  <servlet>
    <servlet-name>DBConnection</servlet-name>
    <servlet-class>DBConnection</servlet-class>

    <init-param>
      <param-name>DBConnString</param-name>
      <param-value>jdbc:oracle:thin:@localhost:1521:WDCT</param-value>
    </init-param>

    <init-param>
      <param-name>UserName</param-name>
      <param-value>wdctuser</param-value>
    </init-param>

    <init-param>
      <param-name>Password</param-name>
      <param-value>pass00word</param-value>
    </init-param>

    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet>
    <servlet-name>ContactInfo</servlet-name>
    <servlet-class>ContactInfo</servlet-class>
    <load-on-startup>2</load-on-startup>
  </servlet>
</web-app>
```



## References

Kuster, E.J (in Preparation) *Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Installation Manual*, DSTO General Document.

Kuster, E.J (in Preparation) *Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Developer Guide*, DSTO General Document.





<b>DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION</b> <b>DOCUMENT CONTROL DATA</b>					
				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	
2. TITLE  Joint Warrior Interoperability Demonstration (JWID) Web Data Collection Tool (WDCT) Installation Guide			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)  Document (U) Title (U) Abstract (U)		
4. AUTHOR(S)  Egon Kuster			5. CORPORATE AUTHOR DSTO Information Sciences Laboratory PO Box 1500 Edinburgh, SA, 5108		
6a. DSTO NUMBER DSTO-GD-0378		6b. AR NUMBER AR-012-921		6c. TYPE OF REPORT General Document	
				7. DOCUMENT DATE September 2003	
8. FILE NUMBER N 9505-23-36	9. TASK NUMBER 01/307	10. TASK SPONSOR DDJWID		11. NO. OF PAGES 32	12. NO. OF REFERENCES 2
13. URL on the World Wide Web  <a href="http://www.dsto.defence.gov.au/corporate/reports/DSTO-GD-0378.pdf">http://www.dsto.defence.gov.au/corporate/reports/DSTO-GD-0378.pdf</a>				14. RELEASE AUTHORITY  Chief, Command and Control Division	
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT  Approved for public release  OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111					
16. DELIBERATE ANNOUNCEMENT  No limitations					
17. CASUAL ANNOUNCEMENT Yes					
18. DEFTEST DESCRIPTORS  Joint Operations, Interoperability, Demonstrations, Data Analysis, Data Recording, Software Tools					
19. ABSTRACT  The Web Data Collection Tool (WDCT) allows for the collection of demonstration assessment information from geographically disperse locations. This installation manual provides the WDCT administrator with instructions on how to install and configure the WDCT software. The WDCT in the past has been used for the assessment of JWID (Joint Warrior Interoperability Demonstration) and the CINC21 (Commander in Chief for the 21st Century) ACTD (Advanced Concept Technology Demonstrator). WDCT is a web-based tool that connects to an Oracle database to store its analysis information and configuration settings. Over the past few years that the WDCT has been used within JWID it has proven to be a valuable tool in eliciting feedback from the warfighters using the demonstrations being assessed.					